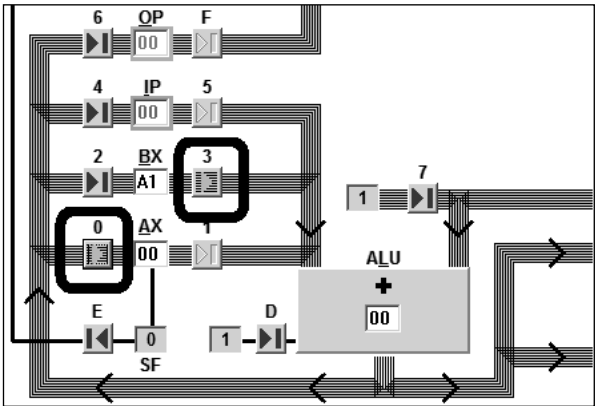
Rechen- und Transportoperationen zwischen Registern

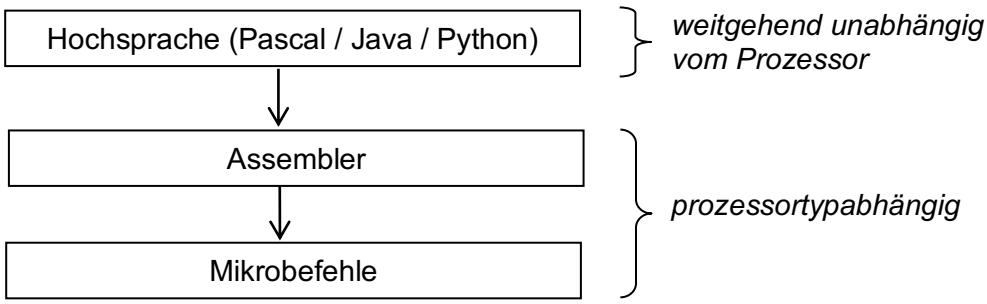
Über das "Öffnen" von Toren können Inhalte eines Registers (8-Bit-Speichers) in ein anderes Register kopiert werden. Der Menüpunkt "Ansicht → Bus ohne Pfeile" wird ausgeklickt, um die *Datenrichtung* der Busse zu sehen.

Beispiel:
Der Wert $A1 steht in Register BX und soll in Register AX kopiert werden. Dazu werden die **Tore 3 und 0** aktiviert und der ***Takt*** gedrückt.



Befehl	AX = BX
Assembler	MOV AX , BX
Mikrobefehl	Tore 0, 3 → Takt

Assembler ist also eine abgekürzte "verbale" Beschreibung eines **Mikrobefehls** (oder mehrerer Mikrobefehle), der das Resultat (Ausführung / Aktion) des Mikrobefehls für einen Menschen verstehbar beschreibt. Assembler ist somit eine Zwischenstufe zwischen *Mikrobefehl* und *Hochsprache*.



Aufgabe 1
Schreibe die Befehle in ***Assembler***-Code um und notiere bei allen Befehlen die Torbelegungen (***Mikrobefehl***). Die Register können nach Belieben mit Startwerten versehen werden.

Bsp: DR = AX | Assembler: MOV DR, AX | Mikrobefehl: Tore 1, A → Takt

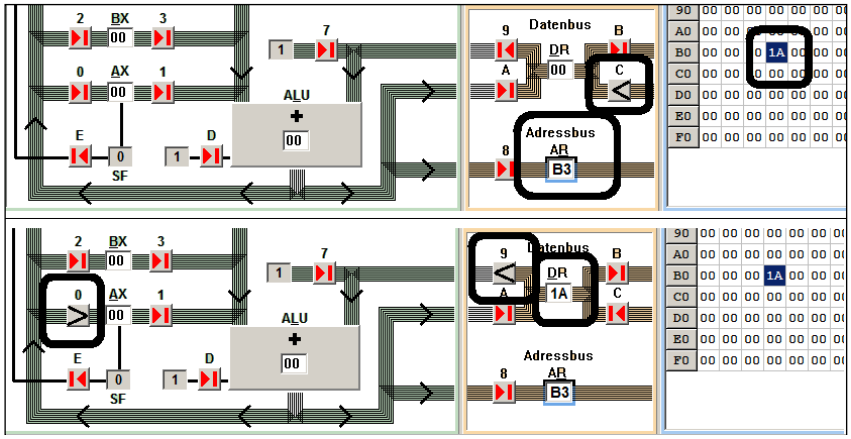
a) AX = DR
b) IP = DR
c) BX = BX + DR (ADD BX, DR)
d) AX = AX - DR (SUB AX, DR)
e) IP = IP + 1

Aufgabe 2
Die folgenden Operationen müssen in mehrere Assembler bzw. Mikrobefehle aufgespalten werden. Zusätzlich wird das Register DR als Zwischenspeicher genutzt.
a) AX = AX + BX
b) AX = AX - BX
c) AX = BX + IP
d) IP = AX – BX

Speicherzugriffe (RAM)

Der Speicher (RAM) ist über den *Datenbus* und den *Adressbus* mit dem Mikroprozessor verbunden. Der Datenverkehr wird ausschließlich über das ***Datenregister DR*** abgewickelt. Das ***Adressregister AR*** enthält die Quell- bzw. Zieladresse der Daten. Zur bequemen Eingabe besitzt MIKROSIM ein RAM-Fenster, das durch direkte Eingabe und über das Dateimenü gefüllt werden kann.

Beispiel: *Die hexadezimale Zahl, die an der RAM-Adresse B3 steht, soll ins Register AX geholt werden.*



Befehl	AX = [B3]	eckige Klammer bezeichnet Adresse, nicht Inhalt
Mikrobefehl	AR = B3 (manuelle Eingabe) Tore C → Takt / Tore 9, 0 → Takt	
Assembler	MOV AX, [B3]	

Aufgabe 3
Schreibe jeweils die **Assembler**- und **Mikrobefehle** auf:

a) BX = [1A]
b) [07] = AX
c) AX = AX + [05]
Assembler: ADD AX, [05]

d) [07] = [05] + [06]
Tipp: Drücke die Zuweisung durch drei Assemblerbefehle aus.
e) [07] = [05] – [06]

Übersicht: **RAM-Adressierungsarten**
- direkte Adressierung MOV AX, [07]
- indirekte Adressierung MOV AX, [BX]
- relative Adressierung MOV AX, [BX+07]